INTRODUCTION

IT departments are always being asked to deliver more for less. Service-Oriented Architecture (SOA) has garnered widespread attention because it promises to do just that – deliver more business benefit while reducing costs.

However, SOA is not just another product or technology to be added to the IT environment. SOA is a significant departure from the traditional large, monolithic applications that currently populate IT environments. SOA is a strategy for constructing business-focused software systems from loosely coupled, interoperable building blocks (called Services) that can be combined and reused quickly, within and between enterprises, to meet business needs.

Of the many benefits that SOA extols - agility, reuse, and integration are generally considered to be the most significant.

- **Agility** – SOA assists with responding more quickly and cost-effectively to changing business needs which, in turn, helps the business respond to market conditions faster.
- **Reuse** – SOA promotes more effective reuse at the macro (business unit of work) level rather than micro (code module or class level).
- **Integration** – SOA simplifies interconnection to, and between, existing IT assets; be it integration with partners and customers, or simply integration of internal applications.

Reaping all of the benefits of SOA requires more than simply learning some new technology and deploying some new products.

Adopting and executing SOA requires strategic thinking and execution. Moving to SOA requires changing the way that IT delivers business solutions across the entire solution delivery lifecycle - from project funding, governance, and requirements management all the way to production monitoring and ongoing maintenance.

Successful SOA adoption requires a holistic approach using proven, pragmatic techniques tailored to the organization’s current capabilities and existing business drivers.

This is why Oracle has developed a pragmatic, holistic approach, based on years of experience with numerous companies to help customers successfully adopt SOA and realize measureable business benefits. See Figure 1 for an overview of Oracle’s approach to SOA.
Oracle’s approach to SOA is a pragmatic, holistic approach, based on years of experience with numerous companies, to help customers successfully adopt SOA and realize measurable business benefits.

Figure 1: Oracle's approach to SOA

There are four essential focus areas which must be addressed to succeed at SOA:

- **Establish a strategic plan for SOA adoption** – A SOA Roadmap provides guidance to the SOA initiative allowing multiple projects to progress in parallel yet remain coordinated and ultimately resulting in a common end goal that provides value greater than the sum of the individual projects.

- **Execute the SOA program level activities** – The program-level efforts create the processes and assets that are leveraged across all of the individual projects. Examples include the service-oriented reference architecture, governance policies and processes, standards, metrics, training and mentoring, service engineering method, etc. The program level efforts provide and enforce the necessary consistency required to succeed at SOA adoption.

- **Deliver projects and services following SOA best practices** - The portfolio of projects is the route to realizing the service-oriented reference architecture, identifying shared services, and delivering measurable business benefit. The projects that make up the portfolio need to be chosen based on their business benefit, risk, and service reuse opportunity. Initial projects drive the service infrastructure build-out and identify the initial shared services. Follow-on projects leverage the service infrastructure and reuse the shared services. Obviously it is the follow-on projects that demonstrate the full value of SOA.

- **Establish ongoing guidance and governance** – Continuous and proactive guidance must be applied to each project team to make sure that
the benefits of SOA are fulfilled. A delicate balance needs to be struck between too little control and too much control. With too little control SOA adoption will be haphazard at best. Too much control may stifle project teams resulting in pushback or, in the worst case, outright defiance.

SOA ADOPTION

To successfully adopt and execute SOA, a company must create a plan that addresses the full extent of the changes required for SOA. Oracle has been working with a wide variety of companies that are in various stages of SOA adoption. This collective experience has been captured in the Oracle SOA Maturity Model such that it can be used to measure the progress of a SOA initiative and, more importantly, can identify specific capabilities that are lacking or lagging and are therefore inhibiting the SOA initiative. A remediation approach for each of the identified inhibitors can be determined from industry best practices and prior experiences. These remedies can then be prioritized and used to create a plan, called the SOA Roadmap, to put the SOA initiative back on track.

Having a SOA Roadmap based on a comprehensive SOA Maturity Model that is constructed using a proven approach and is based on years of collected experience and best practices accelerates the SOA adoption and dramatically reduces the risks associated with the transformation that SOA requires.

Domains

Oracle’s SOA Maturity Model includes over ninety capabilities that capture the best practices that Oracle has collected over many years working with a wide variety of companies. These capabilities provide the detail necessary to truly measure and guide the progress of a SOA initiative. As depicted in Figure 2, there are eight domains in the maturity model:
These eight domains, although interrelated, are sufficiently distinct. To succeed at SOA adoption, an organization must make adequate progress in all of these domains. Inevitably an organization will be more advanced in some domains (and further in some of the capabilities within a domain) than others. Therefore, it is important to be able to measure the relative maturity within each domain (and capabilities therein) and across domains to identify areas that are lagging. Once the lagging areas have been identified it is possible to formulate remedies and thereby improve the success of the overall SOA initiative.

**Maturity**

Within the software industry, maturity is frequently related to the Capability Maturity Model (CMM) and the CMM successor, the Capability Maturity Model Integration (CMMI). Oracle’s SOA Maturity Model parallels this understanding and measures SOA capability against defined maturity levels.

These levels define the path an organization usually takes moving toward SOA maturity. SOA by its very nature requires coordination, cooperation, and a common vision to be successful; therefore, it is necessary to define the strategy before it is possible to be truly successful at repeating it and then ultimately optimizing it.

**Adoption**

Adoption measures how widely SOA is being accepted, embraced, and applied within the enterprise. For smaller organizations within a single line-of-business, maturity and adoption are usually tightly related since there is a single approach to SOA being followed by the entire organization.

However, within large companies with multiple divisions or lines-of-business this is not usually the case. It is common to have one or more divisions that are relatively mature in SOA while other divisions are not even attempting SOA. The SOA Maturity Model handles these situations by providing a separate measure for adoption level. This allows a single division to be effectively evaluated for SOA maturity while still capturing the lack of widespread adoption as a separate measure.

**Measuring Progress**

In order to properly measure the overall progress of SOA initiative in a large organization, the maturity of the individual capabilities and the degree of adoption of such capabilities across the organization is vital. Maturity and adoption levels for some or all of the capabilities or for the domains can be plotted as shown in Figure 3.
Generally, the most effective planning horizon for a SOA Roadmap is 2-3 years. This could be longer or shorter depending on the planning cycles for each organization. The initial phases (e.g. first six months) of the roadmap will contain much greater detail than the later phases. This is appropriate and by design. The SOA journey is a journey of discovery, incremental improvement, and regular course corrections. The SOA Roadmap should be regularly reviewed and updated. The business never stays static, so do not expect the SOA Roadmap to remain static either.

It is important to keep the end goal in mind when applying this roadmap creation process and especially when executing against the roadmap. The end goal is achieving the goal of the SOA initiative. It is NOT to attain a particular score on the SOA Maturity Model.

Refer to the Creating an SOA Roadmap Practitioner Guide at www.oracle.com/goto/ITStrategies for more details.

**SOA INFRASTRUCTURE**

SOA Infrastructure projects differ slightly from the typical SOA project delivery, in that the focus is mostly on enabling infrastructure products, rather than developing new business function, or providing for other business driven needs. These projects play a key role in accelerating adoption of SOA in an enterprise. It allows enterprises to jumpstart their SOA efforts by providing most of the SOA capabilities needed to build and deploy services.

SOA infrastructure should not be a simple installation of products, rather it should be the "realization" of the reference architecture which is aligned with the IT and business strategies. It is recommended to develop functional driven service-oriented reference architecture and a multi-year SOA infrastructure roadmap and
develop the infrastructure gradually. This prudent approach to building infrastructure not only ensures that the IT investment is expended with due diligence but also reduces the risks associated with rolling out SOA in the enterprise.

Some of the challenges and risks that can arise if a service-oriented reference architecture is not utilized are:

- Tightly coupled service connectivity which results in a rigid and inflexible architecture.
- Inconsistent data format and representation of enterprise data entities.
- Misunderstanding which services are available, where they reside, their contract, invocation protocols, and rules for use.
- Unable to monitor and enforce quality of service such as service level agreements described by service contracts.
- Lack of management for service versioning and service life cycle requirements.
- Barriers to the establishment of centralized management of security policies for SOA participants.
- Inflexibility where coarse grained services and applications can’t be composed of existing services.
- Unable to invoke services over heterogeneous transports using varying message brokering capabilities.

Service-Oriented Reference Architecture

The service-oriented reference architecture defines the target architecture and the principles to be used by architects to make architecture and design decisions on their projects.

A service-oriented reference architecture is not just a diagram but contains:

- Multiple views
- Principles and guidelines
- Definitions to aide communication and consistency
- Service classifications
- A set of strategies designed to meet common goals
- Identification of relevant standards
- Best practices and patterns

It is important that the service-oriented reference architecture documents the architecture from multiple views. Each view might include multiple models to illustrate the concepts, capabilities, etc. important for that view. The particular choice of views depends on what material is being covered and which views best convey the information. Example views include conceptual, logical, product
mapping, and deployment views. See Figure 4 for an example high-level conceptual view.

![Service-Oriented Reference Architecture - Conceptual View](image)

**Figure 4: Service-Oriented Reference Architecture - Conceptual View**

The service-oriented reference architecture must be seen as a “living” document whereby incremental releases of the reference architecture will be produced at regular intervals during the execution of the SOA Roadmap.


**SERVICE ENGINEERING**

Many enterprises are still attempting to use traditional delivery methodologies for SOA. However, these methods are designed to deliver projects that do not consider the requirements of the business outside of the scope of the project in question. These methods are too narrowly focused and need to be adjusted to properly enable SOA.

Further, the extensive scope that a SOA may have across an enterprise requires that a repeatable process and sound engineering disciplines be applied through delivery cycles. This is especially true if external or offshore resources are to be utilized. Many enterprises struggle with inconsistency across deliveries for projects that need to coexist. This problem is also an inhibitor to reuse and agility and can be addressed by the adoption of a sound enterprise class service engineering & modeling framework.
Service Engineering and Modeling Framework

An enterprise class service engineering and modeling framework assists with delivering projects within a SOA environment, by not only adding key activities at the project level but also adding key activities such as service identification at the program level.

![Service Engineering Framework Diagram]

Figure 5: Service engineering complements existing project delivery

Figure 5 highlights service engineering tasks that address the engineering requirements not covered by traditional methodologies. Topics covered at the program scope include:

- **SOA Requirements Management** provides a process for harvesting requirements in a manner that naturally facilitates service identification and discovery.

- **Service Identification & Discovery** - establishes the procedures around identifying service candidates, as well as discovering reuse candidates from the existing service catalog. It takes the process from identification and discovery, through the justification processes required to determine if an existing service can be viable for reuse in the proposed manner, or if the proposed service candidate should be realized as a shared service.

- **Service Release Planning** - provides the groundwork necessary for planning for project and service deliveries within a SOA.

Topics covered at the project scope include:

- **Service Definition** - takes the identified service candidates and defines the service boundaries and resulting service contracts.
- **Service Design** - provides the best practices and procedures required to design services from the service contract and produce the service interface.

- **Service Implementation** - provides the guidelines for effectively and efficiently developing shared services.

- **Service Testing** - defines the strategy that should be taken with respect to ensuring the appropriate level of quality for delivered shared services. The information obtained through service testing is also used to enable service deployment.

- **Service Deployment** - defines the guidelines and practices that need to be considered when deploying services into a shared environment.

- **Service OA&M** - covers the operation, administration, and maintenance guidelines required for supporting the SOA's operational environment. OA&M is not simply about keeping the environment operational, but enabling reuse and evolution, in addition to measuring the SOA's adoption and success.

Today, many customers adopting SOA have been addressing some of the issues of delivering SOA projects independently. For instance, there are several different approaches to performing service identification. There are also different approaches to defining the service contract. However, independent solutions to specific problems do little to help the enterprise go from start to finish through the service engineering process. Part of the problem is that a framework or methodology is required that connects these processes.


**SOA GOVERNANCE**

Many companies that have approached SOA via a pilot project have not been seeing the same demonstrated SOA benefits once they have defined a SOA initiative. While pilot projects achieved a level of reuse, they have tended to be within one division, but as soon as a project boundary crosses multiple project teams, additional and new challenges are encountered.

One of the key disciplines to assist in addressing these challenges is SOA governance. While traditional governance has been around a long time, it has been seen as a slow paper-driven process. SOA has heightened the need and importance of having a formal SOA governance model that eases the transition of an enterprise to SOA by providing a means to reduce risk, maintain business alignment, drive a cultural change, and show business value of SOA investments through a combination of people, process, and technology.

It is widely proven that SOA fails to achieve the benefits it promises without a successful governance strategy. The challenge for each enterprise is to determine which SOA governance strategy is best suited to their environment.
SOA Governance Framework

A SOA governance framework helps identify the unique challenges faced by enterprises executing SOA and provides a framework for addressing these challenges. It helps increase the efficiency of the SOA and protect the investments made in SOA assets.

The Oracle SOA Governance Framework complements traditional governance approaches by defining the principles, policies, processes, roles, and infrastructure required to uplift an enterprise’s existing governance strategy.

The Oracle SOA Governance Framework consists of the SOA Governance Reference Model and the SOA Governance Continuous Improvement Loop. The SOA Governance Reference Model is a generic but complete model that is utilized as a baseline SOA governance model to expedite the process of tailoring a SOA governance model for an enterprise. All aspects of the SOA Governance Reference Model are reviewed and considered for customization to the fit the specific enterprise environment.

SOA Governance should be viewed as a program and not a project, therefore the phases of the SOA Governance Continuous Improvement Loop measures progress and updates the SOA governance model when needed to perform any required course correction.
The phases of the Oracle SOA Governance Continuous Improvement Loop include:

- **Assessment**: The framework extols an incremental approach to defining and deploying a SOA governance model, the assessment phase focuses on uncovering the current SOA challenges and prioritizing a SOA governance roadmap.

- **Strategy & Planning**: A major aspect of the SOA governance framework is the definition and future refinement of a customized SOA governance model. This phase defines an appropriate SOA governance model and an associated SOA governance roadmap.

- **Execution**: This phase focuses on the execution of the SOA governance roadmap and managing and enforcing policies while monitoring and ensuring the service levels.

Refer to the *A Framework for SOA Governance* Practitioner Guide at [www.oracle.com/goto/ITStrategies](http://www.oracle.com/goto/ITStrategies) for more details.

**CONCLUSION**

SOA can reduce integration cost, increase asset reuse, increase business agility, and ease regulatory compliance. Achieving these benefits requires a systematic, widespread, holistic, and proficient application of SOA best practices.

Oracle’s approach to SOA addresses the full adoption lifecycle by covering the program and project scopes as well as the governance needs. In addition Oracle’s approach to SOA can be customized and applied against an enterprise’s existing software engineering process or by adopting Oracle’s Unified Method (OUM).

For more information refer to IT Strategies from Oracle. [www.oracle.com/goto/ITStrategies](http://www.oracle.com/goto/ITStrategies)
IT Strategies from Oracle is an authorized library of guidelines and reference architectures that will help you better plan, execute, and manage your enterprise architecture and IT initiatives. The IT Strategies from Oracle library offers two types of best practice documents: practitioner guides containing pragmatic advice and approaches, and reference architectures containing the proven technology patterns to jump-start your initiative.